INVESTEC PROGRAMMABLE BANKING HACKATHON Q4 2021

Russell Knight

BankBot

Automated transfer functionality

TABLE OF CONTENTS

1) INTRODUCTION

1.1 BankBot is a **personal finance solution**, which is housed in a macro enabled MS Excel spreadsheet[1] and makes use of VBA (Visual Basic for Applications) code for automation.

1.2 BankBot makes use of the Investec Programmable banking open **API's**.

1.3 This document **summarises** the information required to assess the solution for purposes of the Q4 2021 **Hackathon**.

2) DECLARATION

2.1 The Hackathon rules expressly forbid the submission of solutions that were **developed prior** to the commencement date.

2.2 It should be noted that the **base solution** (transaction ingestion, categorization etc.) was developed prior to the Hackathon start date, with the newly introduced transfer functionality using it as a foundation.

2.3 Accordingly, the Author understands that only the **transfer** functionality elements are under consideration for purposes of Hackathon evaluation.

3) MARKUP LEGEND IN THIS DOCUMENT

3.1 This form of markup, (braces/curly brackets) indicates an MS Excel Worksheet/Tab: {**Worksheet**}.

3.2 This form of markup (square brackets) indicates a Parameter: [**Parameter**].

4) ANALYSIS OF TRANSFER API APPLICATION

4.1 A solo **brainstorm** was initiated to evaluate the categories of circumstances in which the transfer functionality can be applied to practical business needs.

4.2 Logically, I can see the following **application** categories:

4.2.1 **Interest** optimisation (maximise interest received and minimise interest paid).

4.2.2 Fund identification/**tagging** or ring-fencing of amounts for a specific purpose.

4.2.3 Avoidance of **failed payments** (in the extreme/unlikely condition of the primary account reaching its overdraft limit).

4.2.4 **Silly**/whimsical usage? e.g. using the Descriptor fields to generate a message with multiple transactions such as a multi-line birthday message.

[1] Macro enabled Excel spreadsheets have a file extension of .xls**m** as opposed to .xlsx

4.3 Furthermore – Users can process fixed amount scheduled transfers via **existing** Internet banking functionality. This implies that practical usage of the new API functionality is limited to instances when the transfer requirement/business need is **variable** in terms of either timing, amount, or descriptor (we would not want to create functionality that duplicates existing standard Internet banking abilities).

5) TRANSFER FUNCTIONALITY IMPLEMENTED

5.1 BankBot contains the following functionality which leverages the new Beta access **Transfer API's** released by Investec:

5.1.1 **Manual transfer** functionality at the top of the {Transfers} worksheet, whereby the 5 variables are maintained by the User and the API call is initiated with these variables (primarily to get the API call working at the start and thereafter to set account balances up for purposes of automated transfer solution testing).

5.1.2 **Automated transfer** to maintain the primary Investec account balance within an upper and lower **(optimal) range**.

5.1.3 Automated calculation of a **SinTax**[2] (more detail on this later) and transfer of this amount to a ring-fenced account for distribution by way of donation to worthy causes.

5.2 Maintaining primary Investec account **balance** in the optimal band:

5.2.1 Need for transfer assessment is performed according to the **frequency** set in a scheduler application (such as Windows task scheduler).

5.2.2 Current primary account **balance** is extracted, together with the **available** amount for the account.

5.2.3 The credit facility is a known factor for the primary account – the **pending** transactions are thus able to be deduced.

5.2.4 The primary account balance, reduced by the pending transactions is compared to the **lower limit** for the account [Parameter 19].

5.2.5 If we are on the **last day** of the month or **primary debit orders** have not yet come through the account in the first x days of the month [Parameter 21], then the key payments aggregate [Parameter 20] is added to the target balance to prepare for the debit/stop orders.

5.2.6 The account is **replenished** from a Parameter identified Account ID [Parameter 22][3].

[2] This term is an intended pun centred on the word "syntax" given that the community is comprised of programmers.

[3] For example one's Mortgage account.

5.2.7 Replenishments take place in rounded up **increments** of either ZAR100 or ZAR1000, as per [Parameter 29].

5.2.8 If it is not the last day of the month and it is after [Parameter 21] days in the new month[4], then consideration is had as to whether to **transfer surplus** funds **out** of the primary account (currently set to the same account [Parameter 22]).

5.2.9 Using a **buffer value** [Parameter 25], the **upper** limit for the account is determined.

5.2.10 Taking into account **pending** transactions, surplus funds above the upper limit are **transferred** to the partner account to optimize interest.

5.3 **SinTax** automated transfers

5.3.1 Sintax is **enabled and disabled** via a master switch [Parameter 27].

5.3.2 A **ring-fenced account** (which I renamed to "SinTax account" on my profile) is nominated to receive the transfers [Parameter 28].

5.3.3 SinTax **percentages** are maintained by the User (once-off) for any category that they feel should generate the tax.

5.3.4 With each **API** transaction data ingestion, the following takes place:

5.3.4.1 SinTax value is **calculated** according to the associated percentage derived from the classification of the transaction.

5.3.4.2 The **total SinTax** for the API transaction data ingestion is tallied.

5.3.4.3 The SinTax total is **automatically transferred** to the SinTax account.

5.3.4.4 The User is **notified** of the transfer via rich Telegram messaging.

6) BUSINESS PROBLEMS SOLVED/VALUE ADDITION

6.1 Account balance optimization:

6.1.1 Avoidance of **overdraft**.

6.1.2 Avoidance of exceeding **credit limit** on account and the possibility of declined payments.

6.1.3 **Interest optimization** (either interest paid reduced, or interest received increased).

[4] Most debit orders and stop orders are set for the end of the month or very beginning of the month (though this may need to be tweaked).

6.1.4 User is freed up from having to worry about these elements as they take place **automatically** based on parameters, which can be adjusted according to the changes in circumstances.

6.1.5 **Descriptor** references applied to the transaction contain specifics linked to the parameters that were used to perform the transfer.

6.1.6 Rich Telegram **messaging** keeps the User informed of transfers performed.

6.2 **SinTax** calculation and fund ring-fencing:

6.2.1 **Discourage** unhealthy/self-centred/environmentally unfriendly spend.

6.2.2 Conversely, **feel better** about the spend because it generates funds for charity? (Whichever of these you need)

6.2.3 Add some **science** to the concept of donating to worthy causes, linked to consumption behavior (most Investec account holders will be progressive people).

6.2.4 Flexibility to adjust the **percentage setting** for SinTax determination in the Classification ruleset with ease, according to the User's personal requirements.

6.2.5 Funds are **ringfenced** in a separate account for purposes of donation.

6.2.6 Making a **single** payment to each charity will make it easier to obtain a **S18A** donations tax certificate (as opposed to a monthly stop order).

6.2.7 Contribution to society takes place in an **automated** manner.

7) SETTING UP THE SOLUTION TO EVALUATE

7.1 Copy the **MS Excel file** to a folder on your **local** hard drive and open it in MS Excel (you may get security warnings the first time and need to adjust the Trust settings to accept the spreadsheet as safe).

7.2 Make sure that you are connected to the **Internet**.

7.3 Maintain the following on the **{Parameters}** worksheet:

7.3.1 [Parameter 1] – Set value to “No”[5] if you do not have the necessary **Telegram** Bot information (see later in document for instructions on setting up a Telegram Bot).

7.3.2 Critical - [Parameter 5] - Investec Programmable banking API **Secret**

[5] However, be aware that Telegram messaging is the primary method of BankBot providing feedback to the User. Without this feedback you may find yourself confused as to what has taken place after the code has been run.

7.3.3 Critical - [Parameter 6] - Investec Programmable banking **Client ID**

7.3.4 Critical - [Parameter 9] – Maintain Investec main **account ID**[6]

7.3.5 [Parameter 15] – Choose the **number of days** backward that transactions must be obtained from the transactions API.

7.3.6 [Parameter 19] – Lower **limit** for automated transfer determination

7.3.7 Critical - [Parameter 22] – Maintain the partner **account ID** for automated transfers (Mortgage account for me).

7.3.8 [Parameter 25] – **Buffer** amount (BankBot) will add this to the lower limit to determine the **upper** limit

7.3.9 [Parameter 23] – Transfers **into** main account enabled with a "Yes"[7]

7.3.10 [Parameter 24] – Transfers **out** of main account enabled with a "Yes"[8]

7.3.11 [Parameter 27] – **SinTax** calculation enabled (recommend that you disable at first and then enable it later)

7.3.12 [Parameter 28] – Partner account for SinTax transfers.

7.4 Maintain transaction classifications[9] on the **Classifications** worksheet (I have left all non-sensitive classifications in, to speed this up for you – delete any that you do not want).[10] Basically you should run through your transaction descriptors and maintain the key identification word/s for recurring transactions.

7.5 Generate the **list of bank accounts** in your Portfolio on the {Accounts} worksheet by clicking on the icon (representation below).

7.6 Initiate the **API data collection** and assimilation sequence by clicking on the icon in the {Parameters} worksheet. This will run the primary procedure sequence:

7.6.1 Update the **balances** for each account in the {Portfolio} worksheet.

[6] Note that this is not your traditional bank account number (your account number forms part of the account ID)

[7] Probably better to set this switch off ("No") when the first transaction API call is made.

[8] Probably better to set this switch off ("No") when the first transaction API call is made.

[9] The solution automatically detects the last row that has been maintained to enumerate the rule list, so make sure that you do not place anything but new rules at the bottom.

[10] Don't worry about the numbering column A – it is not used for anything in the code.

7.6.2 Evaluate the primary account balance and perform an **auto-transfer** to bring the balance into desired range, if necessary.

7.6.3 Submit the API call for the **transactions** on the primary bank account and store them on a temporary worksheet {API}.

7.6.4 **Ingest** the API records into the main {Transactions} worksheet

7.6.4.1 **All** transactions on the API sheet will be ingested for the initial call.

7.6.4.2 After the initial call, BankBot will compare the API data to existing transaction data and find the **change point** at which to import to avoid duplication and omission.

7.6.5 **Categorise** each transaction where it can.

7.6.6 Calculate **SinTax** amount on each transaction that belongs to a category that attracts SinTax (if enabled).

7.6.7 Perform an automated **SinTax transfer** (if enabled).

7.6.8 Check for **supporting documents** to associate with the transaction (if enabled via the specification of a folder to monitor)

7.6.9 Notify of **missing** supporting documentation if an expense is expected to have supporting documentation and does not (if specified that the expense should have supporting documentation).

7.6.10 Send a summarized Telegram **message** (if enabled), for each major step in the process.

7.7 Afterward, the User needs to run through the {Transactions} worksheet and **manually categorize** those transactions[11] which were not able to be automatically categorized (they will have the placeholder categorization "ZZZUncategorised").

8) SCHEDULING AUTOMATED RUNS

8.1 Note that scheduling is intended only to take place **after the first/initial ingestion** of transaction data as per the preceding section.

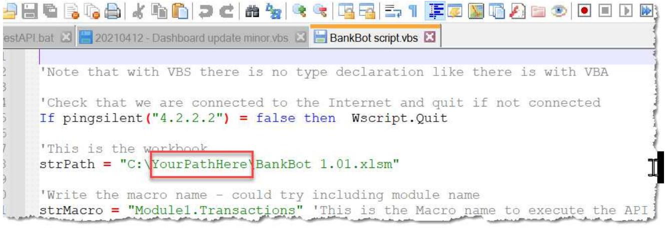
8.2 Reduce the period of transaction query by maintaining **[Parameter 15]** in the {Parameters} worksheet, which informs the procedure how many days retrospectively to collect transactions for (after the initial ingestion, this can be set to 2 days).

8.3 You would already have saved the **BankBot 1.01.xlsm** file in a local drive location that you have full access to.

[11] Note that you can use the native MS Excel filtering functionality in order to filter based on ZZZUncategorised, to restrict the view of transactions to only those that were not classified.

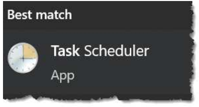
8.4 Save the file **BankBot script.vbs** in a known location (take care, as virus protection often intercepts script files – particularly if you try and place them in sensitive areas such as the root directory).

8.5 Open the file **BankBot script.vbs** with a text editor and adjust the path to point to the location that you have save the file **BankBot 1.01.xlsm**.

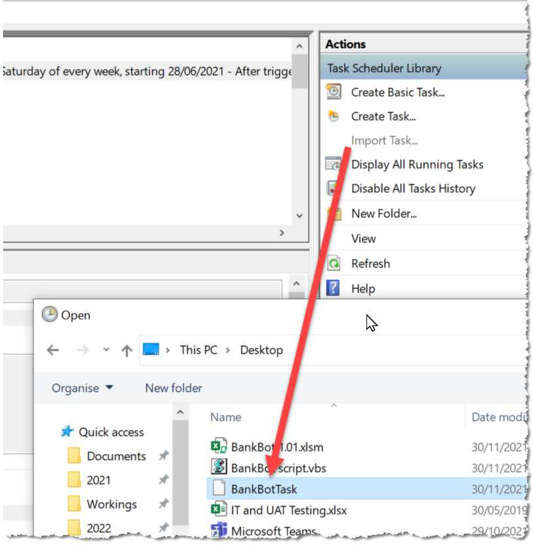


```

'Note that with VBS there is no type declaration like there is with VBA

'Check that we are connected to the Internet and quit if not connected
If pingsilent("4.2.2.2") = false then  Wscript.Quit

'This is the workbook
strPath = "C:\YourPathHere\BankBot 1.01.xlsm"

'Write the macro name - could try including module name
strMacro = "Module1.Transactions" 'This is the Macro name to execute the API
```

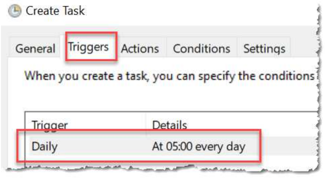
8.6 Open the Windows **Task Scheduler** application.



8.7 **Import** the file BankBot task into Task scheduler (this is just an XML file containing task details).


aturday of every week, starting 28/06/2021 - After trigge
Actions
Task Scheduler Library
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Disable All Tasks History
New Folder...
View
Refresh
Help
Open
This PC › Desktop
Organise
New folder
Quick access
Documents
2021
Workings
2022
Name
Date modi
BankBot 1.01.xlsm 30/11/2021
BankBot script.vbs 30/11/2021
BankBotTask 30/11/2021
IT and UAT Testing.xlsx 30/05/2019
Microsoft Teams 29/10/2021

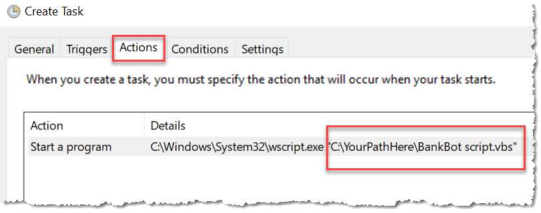
8.8 Adjust the **frequency** or trigger for the automated run:



When you create a task, you can specify the conditions

Trigger	Details
Daily	At 05:00 every day

8.9 Adjust the **path** to the file BankBot script.vbs, to reflect the location that you have saved the script.

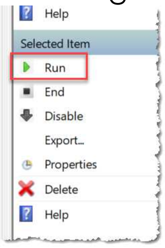


When you create a task, you must specify the action that will occur when your task starts.

Action	Details
Start a program	C:\Windows\System32\wscript.exe "C:\YourPathHere\BankBot script.vbs"

8.10 Click on **OK** to save the task – it should now activate when triggered.

8.11 You can **manually execute** the task to make sure that it runs successfully, by selecting the task and clicking on "Run" in Windows Task Scheduler.



9) ADDITIONAL FUNCTIONALITY/FEATURES/NUANCES

9.1 Automated expense **categorization** and reporting at three levels:

9.1.1 **Category** (top level of categorization/classification)

9.1.1.1 Expense

9.1.1.2 Income

9.1.1.3 Transfer (between accounts)

9.1.1.4 Reclaim (if you have an expense that will later be repaid to you) – want to avoid it distorting your budgeting.

9.1.2 Level **1** – Subcategory e.g. Vehicles

9.1.3 Level **2** – Sub-Sub-Category e.g. Fuel

9.2 Electronic **supporting documentation** association via a link.

9.2.1 BankBot currently analyses only the **file name** itself (not file content[12]) to perform the match.

9.2.2 Filename should be in following **format** to be picked up: **CCYYMMDD – Word**

9.2.3 BankBot will **search** for a transaction in the Transactions worksheet that matches:

[12] Though I may consider exploring this avenue in future to enhance automation.

9.2.3.1 **Transaction date** (not action date, because the supporting document will have the transaction date on it).

9.2.3.2 The **single word** following the hyphen, in the Descriptor/Description of the transaction (keep it short and very specific to the transaction).[13]

9.3 **Budget** monitoring with drilldown capability.

9.4 Usage of **conditional formatting** to:

9.4.1 Highlight **parameter** items that have not been maintained e.g. Telegram messaging on, but no API key for a Telegram Bot.

9.4.2 Highlight the values of **transactions of note** i.e. credit transactions and debit transactions above a certain value.

9.5 Separation of **USD** accounts from ZAR accounts in the Portfolio balances.

9.6 **Pending** transactions are considered when determining auto-transfers.

10) ADDITIONAL INTERPRETATIVE NOTES

10.1 **Transaction** record-based worksheets reflect the latest transaction at the **top**:

10.1.1 {Transactions} worksheet

10.1.2 {Transfers} worksheet

10.2 **Non-transaction** worksheets reflect the last setting/parameter at the **bottom**:

10.2.1 {Parameters} worksheet

10.2.2 {Classification} worksheet

10.2.3 {Accounts} worksheet

10.3 BankBot dynamically detects the **end** of non-transaction worksheets to allow for extension of these worksheets without the need for additional coding. This requires that no text be placed at the bottom of the worksheet other than a new valid record.

11) TECHNICAL ANALYSIS ADVICE

11.1 One can easily **test** transaction API calls and ingestion repeatedly by merely deleting the top x rows on the {Transactions} worksheet and initiating the process again.

11.2 To test balance optimization transfer functionality, one can execute a **manual transfer** to get your balance into the appropriate state by either:

[13] BankBot cannot yet handle association for multiple transactions on the same day with the same vendor (will probably need to have an optional specification of the amount)

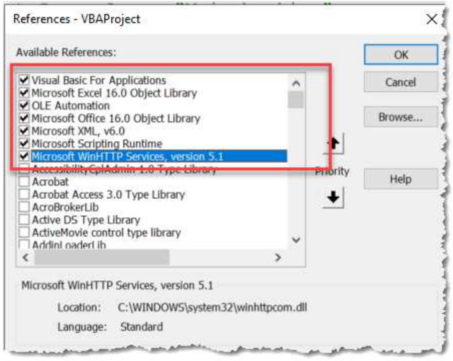
11.2.1 **Regular** Internet banking or the Investec app.

11.2.2 Using the **Manual transfer function** in the {Transfers} worksheet by completing the necessary fields and clicking the button:

⇆

11.3 Pressing **Alt+F11** is the shortcut key to gain access to the underlying VBA coding.

11.4 The following **references to libraries** have been set and are necessary for the executing of the VBA coding (these references should transport with the file, but may depend on the operating system being used):



References - VBAProject

Available References:

☑ Visual Basic For Applications
☑ Microsoft Excel 16.0 Object Library
☑ OLE Automation
☑ Microsoft Office 16.0 Object Library
☑ Microsoft XML, v6.0
☑ Microsoft Scripting Runtime
☑ Microsoft WinHTTP Services, version 5.1
☐ AccessibilityCplAdmin 1.0 Type Library
☐ Acrobat
☐ Acrobat Access 3.0 Type Library
☐ AcroBrokerLib
☐ Active DS Type Library
☐ ActiveMovie control type library
☐ AddinLoaderLib

Microsoft WinHTTP Services, version 5.1

Location: C:\WINDOWS\system32\winhttpcom.dll

Language: Standard

11.5 If wanting to "stretch" the solution with a test that is likely to fail, consider making a **copy** of the MS Excel file beforehand (the file is small and it will allow one to restore to the state prior to the test swiftly).

11.6 Use standard MS Excel **filtering** functionality to analyze treatment of transactions on the {Transactions} worksheet.

11.7 A worksheet with the name {**Launch**} has been set up to provide insight into high level underlying structure of the coding:

11.7.1 Outlining the **main procedures** in the underlying code.

11.7.2 Providing the technical reference to/**name** of the procedure, to allow for it to be easily located in the code.

11.7.3 With **buttons** allowing procedures to be executed in isolation, where this is possible.

11.7.4 Indicating **interdependencies** between the procedures (the main procedure kicks off several other procedures in sequence).

12) CREATING A TELEGRAM BOT

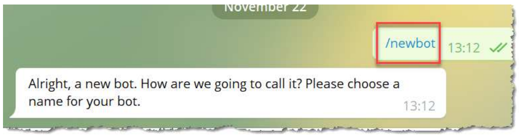
12.1 You need to set up a **Telegram account** by downloading the Telegram application to your mobile phone.

12.2 I recommend downloading the **Telegram application** for your computer and doing this on PC/Mac rather than via a Mobile device.

12.3 Here is a **link** to explain how a new Telegram Bot is created: https://core.telegram.org/bots#6-botfather

12.4 You initiate a chat with the '**BotFather**' Bot – this can take place via the following link: https://t.me/botfather

12.5 Typing the command '**/newbot**' launches the sequence of steps to create your Bot and is very intuitive. Tip – make sure that the username of your Bot is **unique** – most obvious Bot names are already taken.



November 22

/newbot 13:12

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 13:12

12.6 Note that there are **two** names allocated to your Bot:

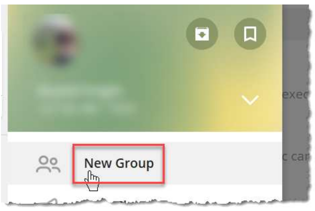
12.6.1 **Friendly** name (this is the first name that is requested and it does not need to be unique).

12.6.2 **Username** – this will have to be unique and it may take a few tries to get one that is.

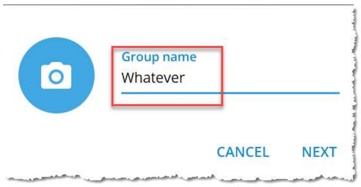
12.7 Make sure you take careful note of the associated **token**, delivered at the end of the process with BotFather, which is a combination of your Bot's ID and an API key, it will look something like this: `110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw`. Take care when copying it – if you miss a character, it will not work.

13) CREATING TELEGRAM BOT CHAT GROUP

13.1 Go to your Telegram menu and create a new **Group**.

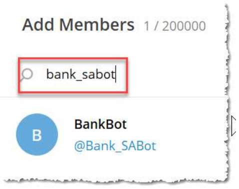


13.2 **Name** the group.



Group name: Whatever

13.3 **Add** the Bot to the group.



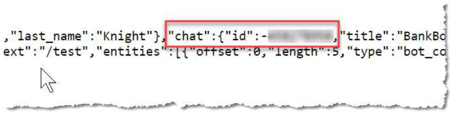
bank_sabot

BankBot
@Bank_SABot

13.4 Send a **message** on the group with any word after a forward slash (forward slash is the predecessor to a Bot command e.g. /Example.

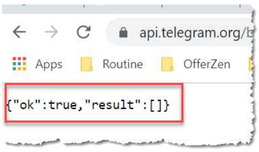
13.5 Put the following address into a web browser (or send a **GET HTTP request**) to get an update from the Bot: https://api.telegram.org/bot*{Yourbottoken}*/getupdates

13.6 The JSON response will contain details of the **Chat ID**. This chat ID will remain static – take note of it for use as one of the key variables later.



```
,"last_name":"Knight"},"chat":{"id":-[redacted],"title":"BankBo
ext":"/test","entities":[{"offset":0,"length":5,"type":"bot_co
```

13.7 If you get **no details** and an empty "result" via the `getupdates` API endpoint, as per the image below, then **remove** your Bot from the group and **repeat** the process.



```
{"ok":true,"result":[]}
```

14) KNOWN SHORTCOMINGS

14.1 **Security** of credentials is inadequate, as this is just a proof of concept.

14.2 The solution has unfortunately **<u>not</u>** been tested on an **Apple** Macintosh as I do not have access to one – I am uncertain how it will behave in this operating system.

14.3 Thorough **error handling** is not in place i.e. the application assumes that key parameters are properly maintained e.g. Account ID.

14.4 The structure of **budgeting** is not currently set up whereby new records can be added easily.

14.5 **Speed** of processing transactions is very slow/inefficient due to the way the dictionary result of JSON parsing is handled (this can be vastly improved) – this is only an issue for the very first ingestion[14], so I have not concerned

[14] When the solution is run daily, the number of incremental transactions it needs to process should be very low compared to the first/initial run.

myself with improving the efficiency/speed of this element for the proof of concept. (Functionality first, perfection later)

14.6 Currently, transactions that are **uncategorized** are treated as an expense from a prudence/inclusion/completeness perspective. Intelligence of this aspect needs to be improved.

14.7 This solution has only been tested in a **Windows** operating system environment – running it in other environments could result in issues arising.

14.8 Currently only acknowledges **USD** foreign currency accounts – all other accounts assumed ZAR.

14.9 The **aesthetics** of the solution need to be enhanced significantly due to “function over presentation” strategy in development.

15) FUTURE ENHANCEMENTS UNDER CONSIDERATION

15.1 Currently trialing automated extraction of **foreign currency exchange rates** to revalue foreign currency denominated accounts (not available in this version).

15.2 Eventually, it is hoped that an API to allow **external EFT’s** will be launched. Automated variable payments could then be initiated (will have to ensure very robust controls to prevent incorrect payments being made).

End of document